Al ejecutar una clase de test que contiene n drivers

a. el método anotado con @BeforeEach se ejecuta n-1 veces
b. el método anotado con @AfterAll se ejecuta n veces
c. el método anotado con @BeforeAll se ejecuta una vez
d. el método anotado con @AfterEach se ejecuta n+1 veces

De los siguientes comandos de maven, ¿cuáles no ejecutarán los tests unitarios?
Comando 1: mvn clean surefire:test
Comando 2: mvn clean test
Comando 3: mvn test
Comando 4: mvn clean compile surefire:test
Comando 5: mvn clean test-compile surefire:test

a. los comandos 2 y 3
b. los comandos 1, 2, 4 y 5
c. los comandos 1, 4 y 5
d. los comandos 1 y 4

Para realizar pruebas de una SUT que contiene dependencias externas,

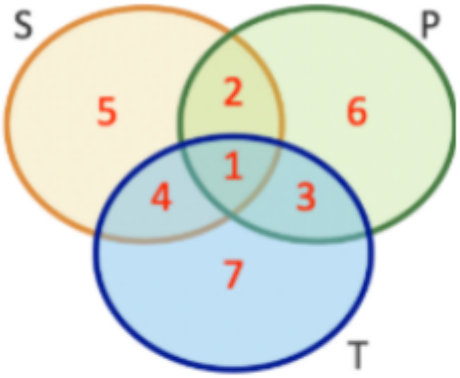
a. usaremos una verificación basada en el estado para pruebas unitarias, y una verificación basada en el comportamiento para pruebas de integración
b. todas las afirmaciones del resto de opciones son falsas
c. el primer paso es identificar los seam que contiene la SUT
d. no está permitido modificar la SUT de ninguna forma para realizar pruebas unitarias

Indica cuál es la complejidad ciclomática del siguiente código:

```
if (a > b && a < c || c < b)
        a = b;
else if (a < b || c > a)
        c = a;
else
        b = c;
```

a. 6
b. 4
c. 3
d. 5

Dado el siguiente diagrama de Venn que hemos trabajado en clase:



Indica cuál de las siguientes afirmaciones es cierta:

a. Un tester debe intentar que el subconjunto 7 sea lo más grande posible
b. Con métodos de caja negra y de caja blanca, se pueden alcanzar comportamientos de los subconjuntos 1 y 2
c. Con un método de caja negra se pueden alcanzar comportamientos del subconjunto 3
d. Con un método de caja blanca se pueden alcanzar comportamientos del subconjunto 4

Cualquier librería que sea requerida en el proceso de construcción de un proyecto Maven

a. si se encuentra en un repositorio remoto, se descarga en el directorio target
b. si no se encuentra en un repositorio remoto, se busca en el repositorio local
c. si no se encuentra en un repositorio remoto, se busca en el directorio .m2
d. todas las afirmaciones anteriores son falsas

Dado el siguiente driver para probar metodoSUT():

```
@Test(expected=MiExcepcion.class)
public void testC1(){
        int resultadoEsperado = 0;
        int resultadoReal = metodoSUT();
        Assert.assertEquals(resultadoEsperado, resultadoReal);
}
```

en el informe de las pruebas para dicho test aparecerá:

a. passed, si metodoSUT() devuelve 0
b. error, si metodoSUT() provoca la excepción MiExcepcion.class
c. failed, si metodoSUT() provoca la excepción MiExcepcion.class
d. passed. si metodoSUT() provoca la excepcion MiExcepcion.class

. Dadas las siguientes particiones de entrada y salida obtenidas al aplicar el método de caja negra de particiones equivalentes, teniendo en cuenta la siguiente codificación para identificar las particiones: 'E' denota entrada; 'S' denota salida; 'V' denota válida; 'nV' denota no válida:
Entrada 1: E1V1, E1V2
Entrada 2: E2V1, E2nV1
Entrada 3: E3V1, E3V2, E3nV1
Salida: S1V1, S1nV1
Si las combinamos de la siguiente manera:
E1V1 - E2V1 - E3V1 - S1V1
E1V2 - E2nV1 - E3V2 - S1nV1

E1V1 - E2V1 - E3nV1 - S1nV1
obtendremos un conjunto de casos de prueba

a. efectivo pero no eficiente
b. eficiente pero no efectivo
c. ni eficiente ni efectivo
d. eficiente y efectivo

Indica cuál de las siguientes afirmaciones es falsa:

a. los métodos de caja negra se pueden aplicar en cualquier nivel de pruebas
b. con métodos de caja negra se pueden probar comportamientos no implementados
c. con métodos de caja negra se pueden diseñar casos de prueba antes de la implementación
d. con métodos de caja negra se pueden probar comportamientos no especificados